2022 2nd International Conference on Computer Science, Electronic Information Engineering and Intelligent Control Technology (CEI)

The six-degree-of-freedom robotic arm optimization iterative learning algorithm improves trajectory planning

HangYu Ji
Changchun University of Science and Technology
Changchun China
1060251630@qq.com

Li Li
Changchun University of Science and Technology
Changchun China
1390376693@qq.com

Abstract—**Among the many fields covered by artificial intelligence (AI), path planning is undoubtedly one of the problems involving a wide range of research areas. Being able to find an optimal solution that allows the robot to establish a safe and effective way to get from the initial state to the final state is a challenge for continuing research today. The increasingly widespread use of robots has made path planning an important aspect of integrating these systems into countless applications. Therefore, taking a 6-DOF manipulator as an example, the mathematical model of working motion of the 6-DOF manipulator was established according to the optimization iterative learning algorithm, and the forward and inverse kinematics were analyzed. In path PLANNING, THE FAST SCALING BASED RANDOM TREE and bidirectional scaling BALANCED connected biTREE path planning method are combined to realize the high-dimensional and complex constraints of non-conflict path planning. The simulation results are in good agreement with the calculation results, which verifies that the structure of the model is consistent with the calculation method, and provides a useful reference for the design of similar manipulator.**

Keywords--six degrees of freedom robotic arm; Optimize iterative learning algorithms; B-Spline; kinematics analysis; Improved trajectory planning

I. INTRODUCTION

With the transformation of the traditional production mode to intelligent production, the application scope of robots is also expanding, and higher requirements are put forward for its control system. However, in the development of industrial robots, there are many problems. Most robot manufacturers develop on their own software interfaces, but their portability is poor; Due to the complexity of the modeling process, a large amount of redundant code needs to be written, which greatly reduces developer productivity. The open source robotics operating system ROS (Robot Operat-ing System) is a cross-platform, multi-language supported, code-reused operating system.

At present, there are three types of motion trajectory optimization for robotic arms: time optimization, energy optimization and hybrid optimization, the most common of which is a robot based on time optimization. In order to achieve the rapid movement of the robot arm and ensure that the trajectory of the robot can reach the target smoothly and accurately, this paper adopts an improved iterative learning method, optimizes the trajectory of the third-order B-spline plan, and proves through experiments that this method can shorten the running time of the path and ensure the stability of the trajectory.

Kinematic analysis of six degrees of freedom robotic arm robotic arm

IRB_120 robot is a six-degree-of-freedom industrial robotic arm developed by ABB, which has the characteristics of small size, high flexibility and compact structure, and is an ideal industrial robot. A physical view of an industrial robotic arm with six degrees of freedom, see Figure 1. Using the optimal iterative method, the kinematics of robots was studied. According to the posture of each joint, the posture and position of the end actuator are obtained by using the homogeneous transformation matrix, which is the so-called kinematic solution; On this basis, using the posture of the end tool, the posture of each joint is determined, This is a kind of inverse solution.

Figure 1. Physical diagram of a six-degree-of-freedom industrial robotic arm

Before establishing the parameter coordinate system, from 0 sequential numbering, according to the D-H method, n degrees of freedom of the robot arm must be set n + 1 parameter coordinate system, in these coordinate systems, the direction of the Z axis and the corresponding hinge axis consistent, the direction of the X axis is located on the two male vertical lines of the connecting rod, because the X axis, Y axis, Z axis are vertical, so you can simplify the parameter coordinate system without the Y axis. As shown in Figure 2, a parametric coordinate system for the corresponding joint is established.

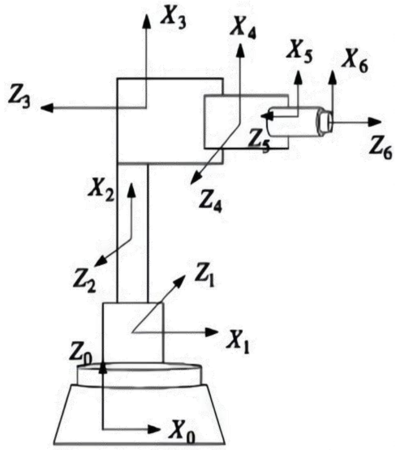


Figure 2. Schematic diagram of the coordinate system of each joint parameter

II. ROBOTIC ARM PATH PLANNING

When the end of the six-degree-of-freedom robotic arm reaches a spatial point, each joint of the robot has multi-solution, and in order to prevent the robot from colliding with the surrounding environment when moving, the path of the robot must be planned. This method uses a rapidly expanding random tree method to randomly sample the state space, so that it can quickly search in high-dimensional space under the condition of complex constraints, and guide it to the airspace, and thus find the planned route from the starting point to the destination.

A. Principle of algorithm

Explore as a tree from the starting location, as shown in Figure 3, where the path is randomly sampled in space until the "branch" covers the target area and the shortest path is selected as the output in the discovered state space. In contrast to the connected double-tree algorithm based on bidirectional scaling equalization, a tree is created in the target region and connected to the first tree until the expansion fails or the first tree is connected.

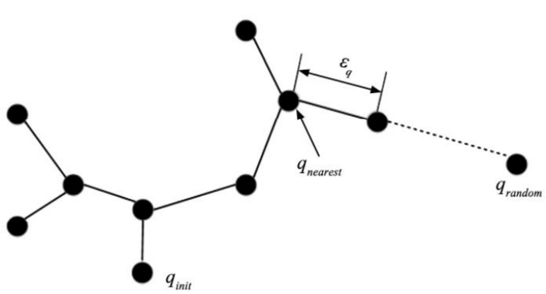
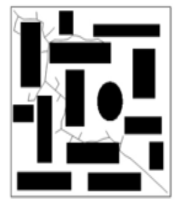
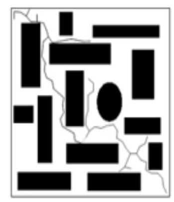


Figure 3. Algorithm schematic diagram

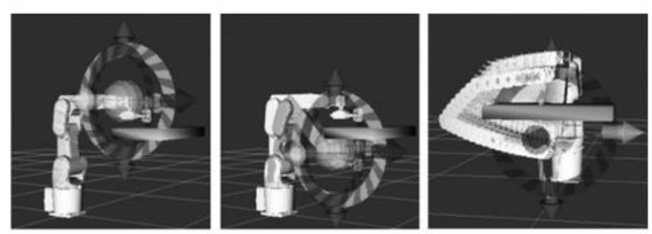
B. Algorithm testing

In the planar route planning, this project uses the MATLAB program, enters a map of 500x500 pixels, defines the starting point source = [10, 10], the target point = [490,490], the iteration step size Stepsize is 20, the maximum number of iterations is 10000 times, and the RRT and RRTConnect algorithms are used for conflict-free path search. Table 1 shows the results of the test.

TABLE 1. COMPARISON OF TEST RESULTS

Type	Random tree	execution time	path length
RRT		0. 871 s	960
RRT Connect		0. 685 s	896

In the spatial path planning test, in order to improve the efficiency of path planning, the RRTConnect algorithm is used instead of the RRT algorithm, and Mo-tionPlanning is inserted into Rviz and Blender is introduced. dae file, add obstacles, set the starting point for moving home and goal; Running the C++ source file to perform the robot collision-free path planning is shown in Figure 4, and the RRTConnect algorithm and RRT algorithm planning took 0.083 s and 0.035 s, respectively.



(a) initial Position (b) Target point Position (c) Path Planning Process

Figure 4. Path planning

The planar and spatial path planning is simulated, and the results show that the method can quickly expand to the other side, so that the expansion of the tree is clearer, the search speed and search efficiency are significantly improved, and it also helps the robot to carry out path planning under complex multi-constraint conditions, so as to achieve the purpose of obstacle avoidance.

III. DESCARTES SPACE ROBOTIC ARM TRAJECTORY PLANNING

Cartesian spatial planning refers to the change of position at the end of the robot arm, when the posture of the robot at a certain point in time is determined, through the inverse kinematics method, the angle of the robot arm can be directly calculated. Its advantage is that the tail trajectory of the robot is easy to observe, and the tail end movement can be more accurately controlled, and can effectively prevent the collision of the robot with space obstacles. However, its disadvantage is that a large number of inverse kinematic operations are performed, and in order to obtain accurate wakes, a large number of intermediate interpolation points must be selected, so the control process is very cumbersome.

In the robot Cartesian space orbit planning, the usual algorithm is: the trajectory of the starting point and the end point of the robot arm determines a path from start to end; By calculating the number of interpolation points and finding several midpoints on that path; Using the endpoint coordinates of a set of robots, change by reversing its six corners. This method is called numerical solution.

A. Speed planning

It can be seen from the mechanical principle that when the robot accelerates while moving, a mechanical collision occurs, which causes wear and tear on the joints. Therefore, when planning the orbit, the speed of each joint must be planned to ensure that the acceleration in motion changes continuously.

The S-type curve is a flow rate curve suitable for this situation, it consists of a parabola and a straight line, and its acceleration curve is continuous. The S-curve is usually divided into three stages, five stages, and seven stages, of which seven stages are based on the continuity of the acceleration, coupled with the constant acceleration of the two ends, making the whole process more stable. The five-segment S-curve is solved according to the hierarchical method, and its curve is represented as follows:

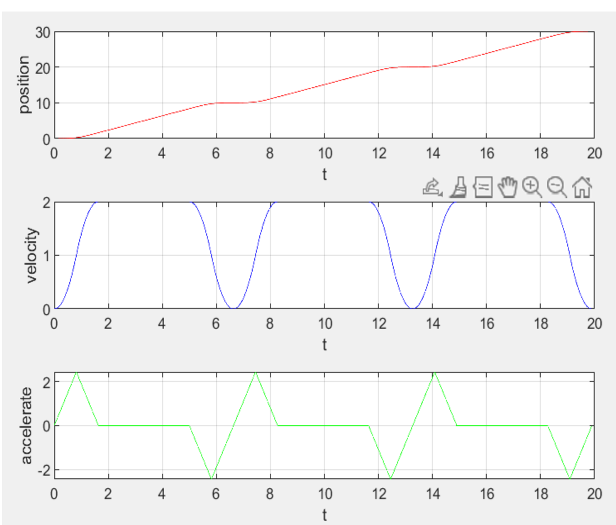


Figure 5. Schematic graph

B. Location planning

Position planning refers to the trajectory planning of the end of the robot arm, usually using a linear interpolation method.

Straight line interpolation is to complete the motion trajectory at the end of the robot arm as a straight line, the operation process is very simple, as long as the starting point and the finish line are connected, and then the number of interpolation points is determined according to the required accuracy, and finally the coordinates of each interpolation point are obtained.

The motion principle of space line interpolation is as follows:

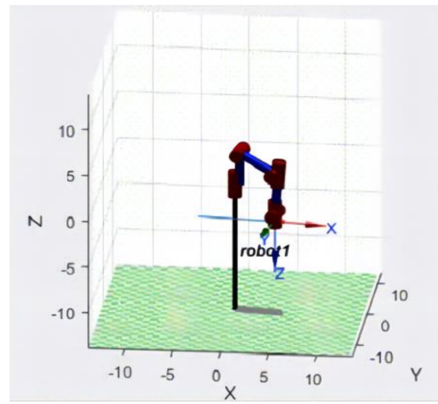


Figure 6. Simulation diagram of spatial linear interpolation motion

In some cases, the position of the end point of some scenes will not change, such as grinding, but when some more complex work needs to be completed, the attitude of the end will also change with movement, such as complex welding processes and so on. When planning postures, it is necessary to take into account the starting and ending positions of the robot arm, and design a posture transformation that meets specific constraints (such as obstacles, work task restrictions, etc.).

From a forward kinematic point of view, the articulation angle θ1 is only related to the transverse coordinates of the manipulator end, so, in the vertical direction, θ1 is constant. In addition, when moving, the position of the end does not change, and it can be seen from the structure of the joint that the joint corners θ4 and θ6 are fixed. From the above analysis, we believe that the vertical movement only needs to examine the changes in the remaining three articulation angles, and its correspondence is as follows.

In this diagram, two angles, a, 2, l, l, are defined, which are fixed at the given start and end points. Their relationship with the articulated angles θ2, θ3, θ5 is as follows:

$$\begin{cases} \theta_2 = \alpha_1 \\ \theta_3 = \frac{\pi}{2} - \theta_2 - \alpha_2 \\ \theta_5 = \frac{\pi}{2} - \alpha_2 \end{cases} \tag{1}$$

$$l_2 \cos\alpha_1 + l_3 \cos\alpha_2 = l$$

On this basis, when planning, only two reverse operations are required to find the starting and ending coordinates of an joint angle, and then the angle of the middle insertion point is found by interpolation, and then a series of angles of other joint angles are obtained according to the interrelationship of the joint angle.

The final simulation results are as follows:

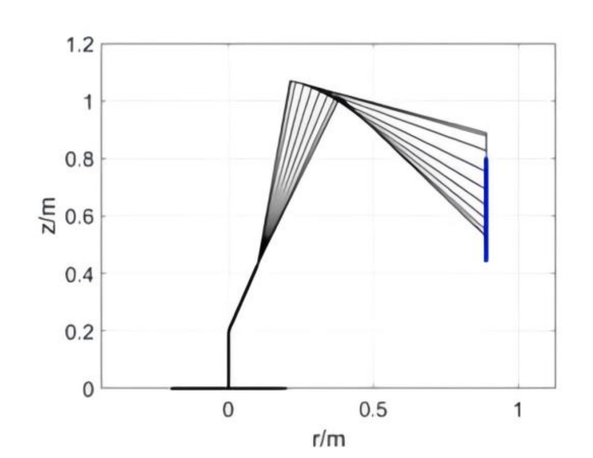


Figure 7. Simulation diagram

The results show that the angular velocity and angular acceleration of each joint change smoothly, and there is no jump point. It makes sense in design.

IV. CONCLUSIONS

In this paper, the trajectory programming problem of the robot arm is studied, and the orbital planning based on the optimal iterative method is adopted, which has better convergence speed and better performance. The results show that the algorithm can find the optimal solution in a shorter time, thereby reducing the running time of the orbit. On this basis, constraints such as torque constraints and transmission constraints can also be added.

Using the orbit planning method, the motion trajectory of the robot end actuator, as well as the angular displacement, angular velocity and angular acceleration of the joints, are obtained. In motion, the angle and time curve of each joint is continuous. In addition, it is possible to effectively limit the velocity and acceleration of the first and second points to 0. It can be seen that in practice, the various joints and moving parts of the manipulator can work smoothly.

REFERENCES

[1] W. Chen, X. Li, H. Ge, L. Wang, and Y. Zhang, "Trajectory planning for spray painting robot based on point cloud slicing technique," Electronics, vol. 9, no. 6, p. 908, 2020.

[2] S. Mahalakshmi, A. Arokiasamy, and J. F. A. Ahamed, "Productivity improvement of an eco friendly warehouse using multi objective optimal robot trajectory planning," Int. J. Prod. Qual. Manag., vol. 27, no. 3, pp. 305-328, 2019.

[3] Y.-H. Wu, Z.-C. Yu, C.-Y. Li, M.-J. He, B. Hua, and Z.-M. Chen, "Reinforcement learning in dual-arm trajectory planning for a free-floating space robot," Aerosp. Sci. Technol., vol. 98, p. 105657, 2020.

[4] V. Sathiya and M. Chinnadurai, "Evolutionary algorithms-based multi-objective optimal mobile robot trajectory planning," Robotica, vol. 37, no. 8, pp. 1363-1382, 2019.

[5] W. Wang, Q. Tao, Y. Cao, X. Wang, and X. Zhang, "Robot time-optimal trajectory planning based on improved cuckoo search algorithm," IEEE Access, vol. 8, pp. 86923-86933, 2020.

[6] S. Zhang, A. M. Zanchettin, R. Villa, and S. Dai, "Real-time trajectory planning based on joint-decoupled optimization in human-robot interaction," Mech. Mach. Theory, vol. 144, p. 103664, 2020.

[7] Á. Madridano, A. Al-Kaff, D. Martín, and A. de la Escalera, "Trajectory planning for multi-robot systems: Methods and applications," Expert Syst. Appl., vol. 173, p. 114660, 2021.

[8] J. Lan, Y. Xie, G. Liu, and M. Cao, "A multi-objective trajectory planning method for collaborative robot," Electronics, vol. 9, no. 5, p. 859, 2020.